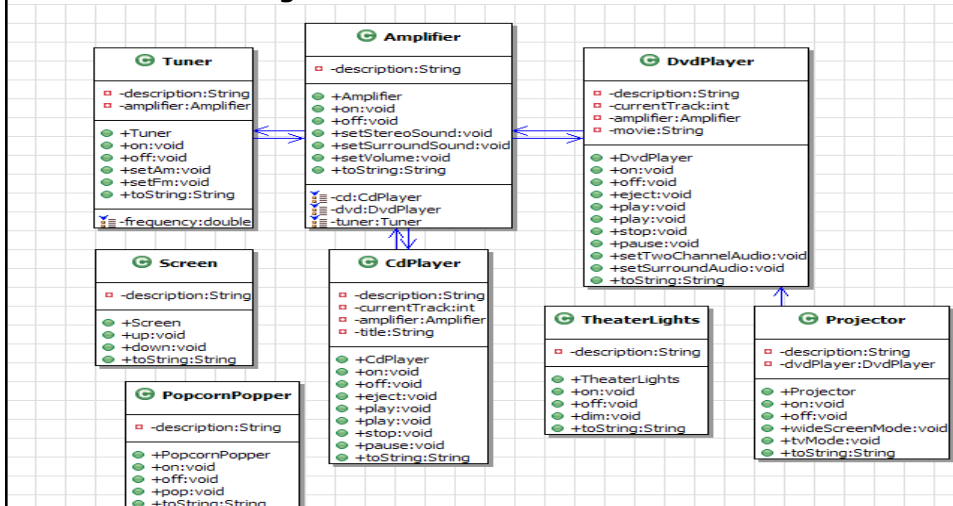


Facade Pattern – structuur van objecten

1

We gaan een pattern bekijken dat de interface verandert, namelijk om de **interface te vereenvoudigen**. Het façade pattern verbergt alle complexiteit van één of meer klassen achter een nette, goed belichte façade.

Voorbeeld: een eigen thuis theater bouwen.



Een film bekijken:

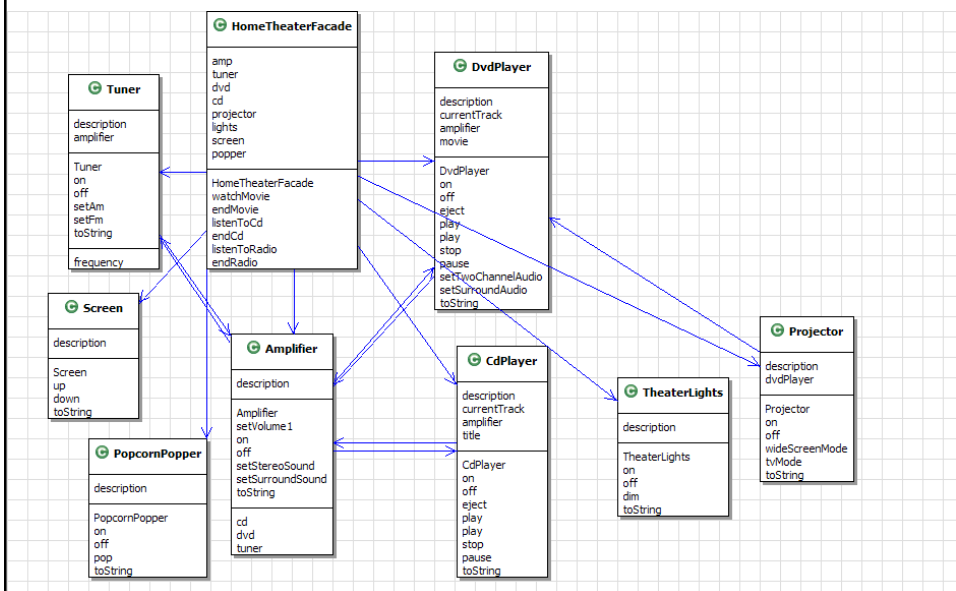
```
//zet de popcornmachine aan en start deze op
popper.on();
popper.pop();
//dim de verlichting naar 10%
lights.dim(10);
//laat het scherm neer
screen.down();
//zet de beamer aan en stel deze in op breedheid voor de film
projector.on();
projector.wideScreenMode();
/*zet de versterker aan, stel deze in op dvd, surround sound en zet
het volume op 5*/
amp.on();
amp.setDvd(dvd);
amp.setSurroundSound();
amp.setVolume(5);
//zet de dvd-speler aan ... en tot slot druk op play!
dvd.on();
dvd.play(movie);
```

GOAD

HomeTheaterFacade

watchMovie()

GOAD



```

public class HomeTheaterFacade {

    private Amplifier amp; private Tuner tuner;
    private DvdPlayer dvd; private CdPlayer cd;
    private Projector projector; private TheaterLights lights;
    private Screen screen; private PopcornPopper popper;

    public HomeTheaterFacade(Amplifier amp,
        Tuner tuner, DvdPlayer dvd, CdPlayer cd,
        Projector projector, Screen screen, TheaterLights lights,
        PopcornPopper popper) {
        this.amp = amp;
        this.tuner = tuner;
        this.dvd = dvd;
        this.cd = cd;
        this.projector = projector;
        this.screen = screen;
        this.lights = lights;
        this.popper = popper;
    }
}

```

```

    public void watchMovie(String movie) {
        System.out.println("Get ready to watch a movie...");
        popper.on(); popper.pop();
        lights.dim(10);
        screen.down();
        projector.on(); projector.wideScreenMode();
        amp.on(); amp.setDvd(dvd);
        amp.setSurroundSound(); amp.setVolume(5);
        dvd.on(); dvd.play(movie);
    }

    public void endMovie() {
        System.out.println("Shutting movie theater down...");
        popper.off();
        lights.on();
        screen.up();
        projector.off();
        amp.off();
        dvd.stop();
        dvd.eject();
        dvd.off();
    } ...
}

```

```

public class HomeTheaterTestDrive {
    public static void main(String[] args) {
        Amplifier amp = new Amplifier("Top-O-Line Amplifier");
        Tuner tuner = new Tuner("Top-O-Line AM/FM Tuner", amp);
        DvdPlayer dvd = new DvdPlayer("Top-O-Line DVD Player", amp);
        CdPlayer cd = new CdPlayer("Top-O-Line CD Player", amp);
        Projector projector = new Projector("Top-O-Line Projector", dvd);
        TheaterLights lights = new TheaterLights("Theater Ceiling Lights");
        Screen screen = new Screen("Theater Screen");
        PopcornPopper popper = new PopcornPopper("Popcorn Popper");

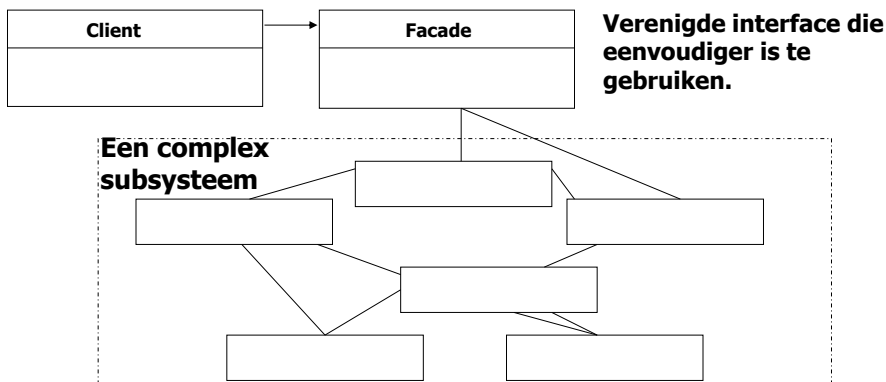
        HomeTheaterFacade homeTheater =
            new HomeTheaterFacade(amp, tuner, dvd, cd,
                                projector, screen, lights, popper);
        homeTheater.watchMovie("Raiders of the Lost Ark");
        homeTheater.endMovie();
    }
}

```

Get ready to watch a movie...
 Popcorn Popper on
 Popcorn Popper popping popcorn!
 Theater Ceiling Lights dimming to 10%
 Theater Screen going down
 Top-O-Line Projector on
 Top-O-Line Projector in widescreen mode (16x9 aspect ratio)
 Top-O-Line Amplifier on
 Top-O-Line Amplifier setting DVD player to Top-O-Line DVD Player
 Top-O-Line Amplifier surround sound on (5 speakers, 1 subwoofer)
 Top-O-Line Amplifier setting volume to 5
 Top-O-Line DVD Player on
 Top-O-Line DVD Player playing "Raiders of the Lost Ark"
 Shutting movie theater down...
 Popcorn Popper off
 Theater Ceiling Lights on
 Theater Screen going up
 Top-O-Line Projector off
 Top-O-Line Amplifier off
 Top-O-Line DVD Player stopped "Raiders of the Lost Ark"
 Top-O-Line DVD Player eject
 Top-O-Line DVD Player off

Facade Pattern

Het **Facade Pattern** zorgt voor een vereenvoudigde interface naar een verzameling interfaces in een subsysteem. De façade definieert een interface op een hoger niveau zodat het gebruik van het subsysteem vereenvoudigt.





Het Principe van de Kennisabstractie



Praat alleen met je directe vrienden. Hoe minder je weet hoe beter.

Dit principe voorkomt dat we ontwerpen maken waarbij veel klassen gekoppeld worden, zodat veranderingen in het ene deel van het systeem invloed hebben op andere delen. Wanneer je veel afhankelijkheden aanbrengt tussen klassen, dan maak je een teer systeem dat duur in onderhoud is en bovendien voor anderen moeilijk te begrijpen.

9

De volgende klasse Car demonstreert alle manieren waarop je methoden kunt aanroepen, terwijl je toch aan het Principe van Kennisabstractie houdt:

```
public class Car
{
```

```
    private Engine engine;
    //andere instantievariabelen
```

```
    public Car()
    { //initialiseer motor, etc.
    }
```

```
    public void start(Key key)
    {
```

```
        Doors doors = new Doors();
        boolean authorized = key.turns();
        if (authorized)
        {
            engine.start();
            updateDashboardDisplay();
            doors.lock();
        }
    }
```

```
    } ...
}
```

Hier creëren we een nieuw object, de methoden zijn legaal.

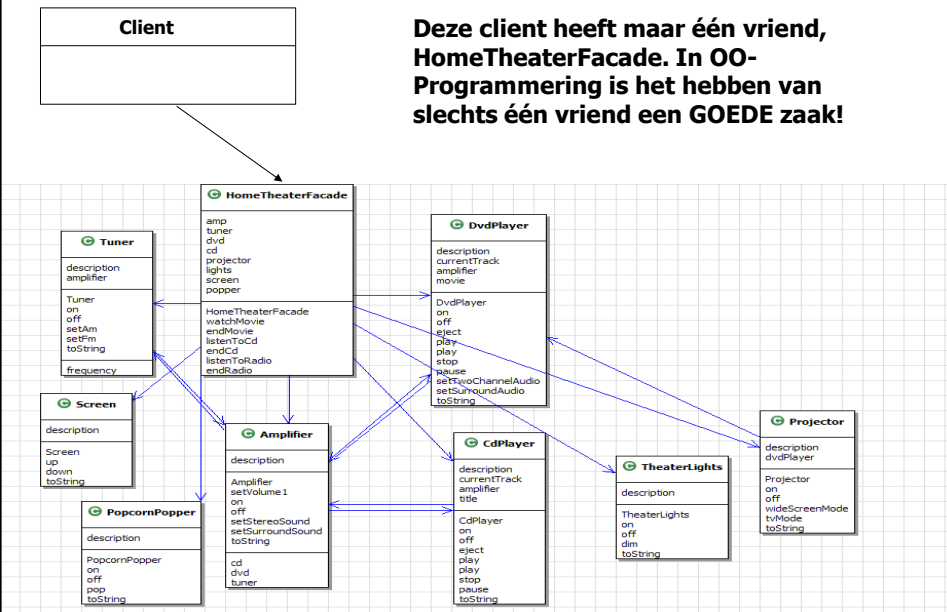
Je kunt een methode voor een object aanroepen die als parameter is doorgegeven.

Je kunt een lokale methode binnen het object aanroepen.

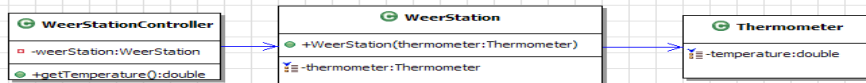
Je kunt een methode van een object dat je gemaakt of geïnstantieerd hebt.

De Facade en het Principe van de Kennisabstractie

OOD



OEFENING: Pas het principe van Kennisabstractie toe



```

public class WeerStationController {
    private WeerStation weerStation;
    public double getTemperature()
    {
        Thermometer thermometer = weerStation.getThermometer();
        return thermometer.getTemperature();
    } //OF
    //return weerStation.getThermometer().getTemperature();
} ...

public class WeerStation
{
    private Thermometer thermometer;
    public Thermometer getThermometer() {
        return thermometer;
    } ...
}

public class Thermometer
{
    private double temperature;
    public double getTemperature() {
        return temperature;
    } ...
}
    
```